iDo



Think different.

I mean, at least I do it by myself

شماره چهارم

قبیله گیک ها



تنها مجله مخصوص گیک های ایرانی

لینوکس (یک)

called software simply started computer short start create systems Torvalds programmers individuals Linus kernel free basic make lets Debian Project thousands use run association supported set cause utilities core common created operating Linux housekeeping program made GNU/Linux currently fundamental system piece worldwide programs



Linux

فهرست

سخن سردبیر

قبیله گیک ها چیست؟ اعضای آن چه کسانی هستند؟ هدف از ایجاد این قبیله چیست؟

اینها سوالاتی است که در تمام مدتی که از انتشار مجله قبیله گیک ها میگذرد از ما پرسیده می‌شود و به همین دلیل تصمیم گرفتیم در اولین مطلب این شماره به این سؤالات تا حدودی پاسخ دهیم. قبیله گیک ها درواقع اجتماعی از دوستان با تفکرات گیکی است که در محیط اینترنت به دور هم جمع شده و سعی میکنند تعریفی مدرن و بهینه از کلمه گیک را بوجود بیاورند. در قبیله ما نظرات میتوانند متفاوت باشند اما نباید مانع رشد قبیله شوند. این یعنی من و دیگر اعضای قبیله میتوانیم با یکدیگر در نوع تفکر راجع به هر موضوعی اختلاف نظر داشته باشیم اما این اختلاف نظر نباید تبدیل به ناراحتی طرفین شود. میتوانیم در مطالبمان هرگونه انتقادی به عقاید یکدیگر انجام دهیم اما نباید این انتقادات از سر نفرت و یا در جهت تخریب شخصیتی باشد بلکه واقعا باید مانند یک دوست انتقاد کرد و بدنبال بزرگ کردن شخصیت خود نباشیم بلکه به دوست هم قبیله ای خود کمک کنیم. ما همه در قبیله با یکدیگر دوست هستیم اما الزاماً با یکدیگر همفکر نیستیم ما این دو موضوع را از هم تفکیک میکنیم و همگی سعی در درک درست این موضوع داریم.

ما در این قبیله هر ماه بر روی یک موضوع خاص تمرکز میکنیم و سعی میکنیم با همکاری یکدیگر بخش های جالب و با ارزش آن موضوع را پیدا کرده، فرا بگیریم و نتیجه آن را بصورت یک متن یک یا دو صفحه‌ای در مجله قبیله گیک ها منتشر نماییم. لطفا توجه داشته باشید که ما برای اشخاص معمولی نمینویسیم مخاطب ما گیک‌ها هستند افرادی مثل خودمان که اگر چیزی برایشان جالب باشد جستجو میکنند، تحقیق میکنند و فرا میگیرند. ما مطالب آموزشی نمینویسیم، ترجمه نمیکنیم زیرا این موارد به اندازه کافی دراینترنت موجود میباشند، بلکه مانند هر گیک دیگری ما مطالب متنوع را از منابع مختلف مطالعه میکنیم و بهترین و جالب ترین قسمتها را انتخاب میکنیم و با استفاده از درک خودمان از آن مطالب یک متن جدید تولید میکنیم. ما مطالبی را می نویسیم که برای گیک ها جالب باشد تا اگر به آن موضوع علاقه داشتند برای یادگیری بیشتر در مورد آن جستجو کنند به همین دلیل ما سعی می کنیم که مطالب کوتاه، مختصر و مفید باشد تا مخاطب را از خواندن دلسرد نکند و همچنین باعث به وجود آمدن جرقه اولیه در ذهن آنها شود. اینکه چه مینویسیم بستگی به شخص نویسنده دارد ممکن است در زمینه سخت افزار یا نرم افزار باشد، ممکن است در مورد یک ایده جدید باشد و یا یک محصول جدید خلاصه مواردی از این دست که بستگی به نویسنده آن مطلب دارد.

ما نتیجه این دوستیهایمان را در قالب یک فایل به نام مجله قبیله گیک ها منتشر میکنیم.

شما نیز میتوانید به قبیله ما بپیوندید، فقط کافی است با **@GeeksTribeMagz** تماس بگیرید.

(@BoBzBoBz)

به یاد یک نابغه

انقلاب لینوکس در دنیای IT خیلی گسترده بود و با قابلیت هایی که این سیستم عامل در اختیار کاربران قرار می داد به سرعت محبوبیت خاصی در اجتماع کاربران کامپیوتری پیدا کرد و دلیل آن متن باز بودن سیستم عامل بود که به کاربران اجازه میداد آن را ویرایش کرده و توزیع مناسب برای کار و پروژه خود ایجاد کنند.

در این میان یکی از مهمترین توزیع های ساخته شده "دبیان" می باشد که در این مطلب به معرفی طراح و سازنده آن می پردازیم.



همان طور که می دانید دبیان یک توزیع موفق است که در سال ۱۹۹۶ به وسیله یان مورداک ساخته و پایگذاری شد، به همین دلیل یان مورداک را پدر دبیان می نامند که با ساخت این توزیع بزگترین تحول را در دنیای لینوکس به وجود آورده است.

یان مورداک متولد ۲۸ آوریـل ۱۹۷۳ و در آلمان چشم به جهان گشود.

او دبیان را وقتی دانشجوی Purdue University مقطع کارشناسی در رشته علوم کامپیوتر بود، در سال ۱۹۹۶ نوشت.

نام این توزیع از اولِ اسم همسرش که Debra Lynn است و اولِ اسم خودش Ian، ساخته شده است.

یان مورداک در سال ۲۰۰۶ به سِمَتِ مدیر بخش تکنولوژی گروه استانداردهای آزاد منتسب شد و همچنین به عنوان عضو اصلی گروه استاندارد پایه لینوکس انتخاب شد. او این کار را با عنوان CTO تا وقتی ادامه داد که بنیاد لینوکس و گروه استانداردهای آزاد، شکل تازه ای به خود گرفت، این دو با هم ادغام شدند و آزمایشگاه توسعه متن باز را ایجاد کردند.

در مارچ ۲۰۰۷ یان از بنیاد لینوکس خارج شد و به شرکت sun microsystem پیوست، در آنجا سیستم عامل سولاریس را پایه گذاری کرد که از لینوکس ساخته شده است و نام کامل این توزیع open Solaris است که دارای محیط کاری گنوم و یک سیستم مدیریتی تحت شبکه می باشد. از مارچ ۲۰۰۷ تا فوریه ۲۰۱۰ به عنوان معاون شرکت Emerging Platforms at Sun مشغول به کار بود تا وقتی که این شرکت با شرکت بزرگ Oracle ادغام شد و او از سِمَتِ خود در این شرکت استعفا داد.

او از سال ۲۰۱۱ تا سال ۲۰۱۵ معاون جامعه پلتفرم و توسعه دهندگان در Salesforce Marketing Cloud که در ایندایاناپلیس(Indianapolis) است مشغول به کار شد و بعد از آن از نوامبر ۲۰۱۵ تا ۲۸ دسامبر ۲۰۱۵ در docker مشغول به کار بود.



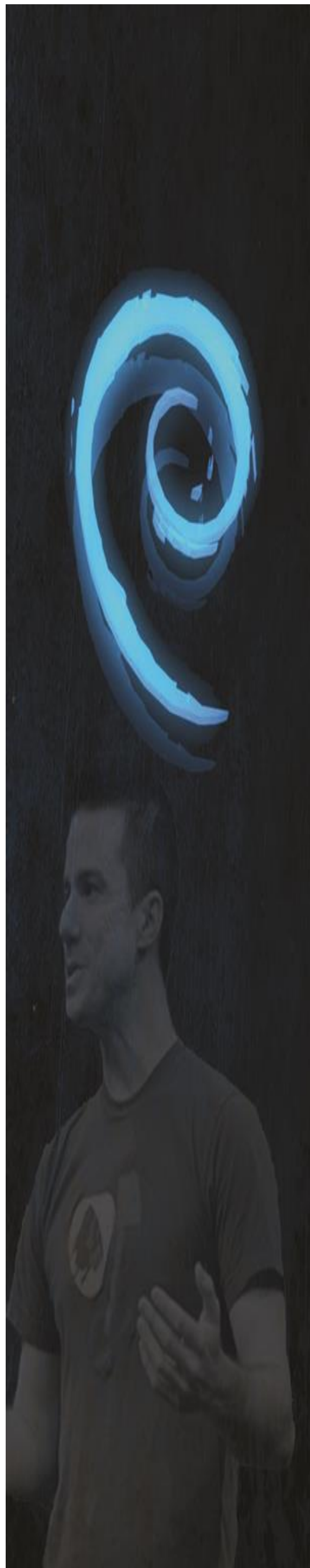
یان مورداک با ساختن این توزیع امکانی را فراهم کرد که با زدن دستور apt-get برنامه ها نصب میشوند و اگر برنامه وابسته ای هم نیاز باشد به همراه برنامه اصلی نصب خواهد شد.

پیش تر اگر قصد نصب برنامه ای را داشتید وقت زیادی صرف می شد و همچنین لازم بود هزینه های گزافی را به شرکت ها پرداخت کنید تا موفق به نصب برنامه مورد نیازتان شوید، اما حالا فقط با دستور apt-get install هر برنامه ای که بخواهید را می توانید به راحتی نصب کنید.

دبیان و ابزار apt-get به دلایل فنی به موفقیت بزرگی دست پیدا کردند، دبیان هم مانند لینوکس به طور مداوم به وسیله اجتماع برنامه نویسان و توسعه دهندگان در حال پیشرفت می باشد و می توانند نظرات خود را برای بهتر شدن به اشتراک بگذارند.

یان مورداک در ۴۲ سالگی در تاریخ ۲۸ دسامبر ۲۰۱۵ دیده از جهان فرو بست.

(@Badrinex)



جنگ قدرت‌ها



اگر بخواهیم لینوکس و ویندوز را در هر زمینه‌ای با هم مقایسه کنیم و بگوییم کدام سیستم عامل کاربردی‌تر است، مطمئنا به پاسخ قاطعی نخواهیم رسید؛ زیرا طرفداران هرکدام از این سیستم عامل‌ها، دلایل خاص و منطقی خودشان را برای حمایت از سیستم عامل محبوبشان دارند و در نهایت به این نتیجه می‌رسیم با توجه به فعالیتی که از سیستم عامل انتظار داریم و با در نظر گرفتن امکانات موجود، یکی از این دو را انتخاب کنیم. ما در این مطلب قصد داریم در مورد مزیت های سرورهای لینوکسی نسبت به سرورهای ویندوزی صحبت کنیم و به نقاط ضعف و قدرت هر کدام بپردازیم.

پایداری و استحکام :

سرورهای لینوکسی به خوبی توانایی اجرایی بودن در سال‌های پیاپی بدون هیچ توقف و شکست را دارند. در واقع، بسیاری از کاربران لینوکس هیچ وقت با یک توقف و خرابی در سرور مواجه نمیشوند، و این ویژگی از اهمیت بالایی برای کاربران برخوردار است، خصوصا برای کسب و کارهای حساس که از کار افتادگی سرور می تواند عواقب فاجعه بار برای آنان داشته باشد لینوکس خیلی بهتر از ویندوز می تواند تعداد زیادی از فرآیندها را در یک لحظه اجرا کند، در وب هاستینگ لینوکس نیازی به راه اندازی مجدد نمی باشد در حالی که در تغییرات تنظیمات ویندوز به طور معمول نیاز به راه اندازی مجدد باعث از کار افتادگی اجتناب ناپذیر می شود . در لینوکس عدم احتیاج به راه اندازی مجدد تضمین شده است . تقریبا تمام تغییرات پیکربندی لینوکس می تواند صورت بگیرد در حالی که سیستم در حال اجرا و ارائه خدمات نامربوط است. اما چه عواملی استحکام و پایداری سرورهای لینوکسی را تا این حد افزایش می‌دهد؟

پاسخ به این سوال را باید در معماری سیستم های لینوکسی و ویندوزی یافت.تفاوت اصلی لینوکس و ویندوز در هسته‌ی آن‌ها است که در لینوکس هسته اصلی را کرنل و در ویندوز رجیستری می دانیم. بعد از نصب هر برنامه در ویندوز تنظیمات آن برنامه در رجیستری ویندوز اعمال می‌شود و برنامه در هنگام اجرا به رجیستری میرود و از آنجا فایل های مربوط به خودش را باز می‌کند و این یعنی اگر برنامه‌ای هنگ کند، رجیستری ویندوز هنگ می‌کند و به عبارتی کل سیستم از کار میافتد و یا اگر ویروسی وارد سیستم شود، ابتدا رجیستری مورد حمله قرار می‌گیرد، زیرا آسیب پذیرتر است و هم چنین مرکز اصلی ویندوز می‌باشد. در لینوکس روند متفاوت است و کرنل وظیفه‌ی ارتباط بین برنامه ها و سخت افزارها را بر عهده دارد.

وقتی برنامه ای را در لینوکس نصب می کنیم، تغییری روی کرنل لینوکس اعمال نمی‌شود و تنها یک فایل در کنار نرم افزار ساخته می‌شود و تنظیمات آن برنامه درون فایل ذخیره می‌گردد. اگر برنامه ای در لینوکس خراب شود، دلیل بر خرابی کل لینوکس نیست و به راحتی می‌توان آن مشکل را برطرف کرد. اگر در لینوکس برنامه ای هنگ کند، هسته سیستم هنگ نمی‌کند و اگر محیط گرافیکی را درگیر نکند، به سادگی می‌توان آن برنامه را بست و دوباره باز کرد.

امنیت :

ویروسها، تروجانها، جاسوس ابزارها، تبلیغات و.... همگی موارد مخربی هستندکه باعث آسیب به سیستم های کامپیوتری و اطلاعات موجود در آنها می‌شود که متاسفانه ویندوز به راحتی به آنها اجازه ورود می‌دهد. بنابراین ما در ویندوز به موارد زیر نیاز داریم :

۱. Firewall
۲. Antivirus
۳. برنامه‌های ضد تبلیغات

که تهیه این موارد در ویندوز مستلزم صرف هزینه می‌باشد.

در نرم‌افزارهای متن باز بر طبق لایسنس GPL که مجوز انتشار لینوکس است، شما می‌توانید کد های یک برنامه را آزاد دریافت کنید و آن را ویرایش کرده و تحت همان لایسنس آن را منتشر نمائید. هر برنامه نویس در هر جای کره زمین، به کدها دسترسی دارد و می‌تواند آن را مرور نماید و از لحاظ امنیتی آن را مورد بررسی قرار دهد. اگر در یک نرم‌افزار متن باز یک حفره امنیتی کشف شود، هرکسی در جامعه نرم‌افزارهای متن باز می‌تواند این مشکل را مورد بررسی قرار دهد و آن را برطرف نماید و مشکلات ظرف چند روز و حتی در برخی موارد ظرف چند ساعت حل می‌گردند، بنابراین سرعت رفع مشکلات امنیتی در لینوکس بسیار بالا است، در حالی که در ویندوز کاربران هیچ نوع دسترسی به کدهای اصلی سیستم عامل ندارند و در صورت کشف یک مورد امنیتی، کاربران فقط می‌توانند آن را گزارش دهند . در دنیای گنو/لینوکس، کاربر آزاد است تا توزیع مورد علاقه خود را از میان تعداد زیادی توزیع موجود، انتخاب کند. هر توزیع نیز با ویژگی‌های امنیتی مختص خود ارائه می‌شود، برای مثال Fedora از SELinux استفاده می‌کند و Ubuntu از AppArmor و هر کدام از توزیع‌ها نیز این برنامه‌ها را به سلیقه خود تنظیم کرده‌اند. در نتیجه عملکرد بدافزار بسیار محدود شده و پس از مدتی از بین خواهد رفت. در ویندوز گاهی در هنگام نصب بسیاری از برنامه‌ها، ما مجوز دسترسی‌هایی را می‌دهیم که نرم‌افزار می‌تواند به بخش‌های مختلف سیستم ما دسترسی داشته باشد و بسیاری از تنظیمات را تغییر دهد که این موضوع می‌تواند خطرات جبران‌ناپذیری را به سیستم ما وارد کند. اما لینوکس از یک سیستم مجوزدهی هوشمند استفاده می‌کند،که هر کار سیستمی ای که بخواهیم انجام دهیم، از ما Password گرفته میشود که اگر آنرا نداشته باشیم، از ادامه کار جلوگیری می‌کند و با این کار دیگر ویروس ها نمی‌توانند دستکاری در سیستم انجام دهند، چون مجوز این کار را ندارند.

- نحوه کار این سیستم هوشمند به صورت زیر است :

۱. سیستم عامل ابتدا نوع فایل را از لحاظ اجرایی بررسی می‌کند.
۲. مجوز کاربر برای اجرا بررسی می‌شود.

۳. در صورت تایید دو مورد بالا فایل اجرا می‌شود و منابع سخت افزاری در اختیار فایل قرار می‌گیرند.

سخت افزار:

سرور ویندوز به طور معمول نیاز مکرر به ارتقاء سخت افزاری دارد، درصورتی که لینوکس انعطاف پذیرتر و مقیاس پذیرتر است. در لینوکس می توان هنگام نیاز به خدمات خاص، به راحتی پیکربندی مجدد انجام داد، در نتیجه این باعث کاهش احتیاج به حافظه، بهبود عملکرد و بالا بردن کارایی سیستم می‌شود.

TCO (Total Cost of Ownership)

لینوکس هیچ هزینه‌ای ندارد و این نرم‌افزار به صورت کامل Free می‌باشد، حتی نسخه سازمانی آن که با حمایت شرکت های بزرگ خریداری شده‌است، به کلی از ویندوز یا نرم‌افزارهای اختصاصی که معمولا شامل هزینه خرید لایسنس می‌باشند، ارزان‌تر است.

لینوکس خصوصا برای امنیت ارزان‌تر از ویندوز می‌باشد.

راحتی کاربرد :

با استفاده از لینوکس، هیچ محدودیتی برای بدست آوردن محصولات خاصی که مد نظر ما باشد، وجود ندارد و ما می‌توانیم به راحتی به انتخاب بهترین راه حل برای کسب و کار خود باشیم. به طور خلاصه با وجود همه مزیت هایی که لینوکس در عرصه سرور فراهم می‌کند، هیچ جای تعجبی ندارد که بسیاری از دولت‌ها، سازمان‌ها و شرکت های بزرگ در سراسر جهان از جمله آمازون و گوگل، از سیستم عامل‌های متن باز در سیستم‌های خود استفاده می‌کنند.

اما ذکر چند نکته در سرورهای لینوکسی نیاز است :

۱. سرور های هاستینگ لینوکس، بسیاری از زبان های مختلف وب را همانند PHP ساپورت می‌کنند. ولی یکی از اشکالات سرورهای لینوکس این است که از زبانهای ASP & ASP.NET که از فناوری‌های شرکت مایکروسافت هستند و پایگاه داده MSSQL پشتیبانی نمی‌کند.

۲. در خصوص پایگاه داده روی سرورها، SQL SERVER مخصوص سرور های ویندوزی و mysql مخصوص سرورهای لینوکسی هستند. در خصوص مقایسه این دو پایگاه داده نیاز به کارشناسی توسط نیروهای متخصص است که در این مطلب نمی‌گنجد، ولی در کل هر دو دارای کارائی های مشابهی می‌باشند.

۳. در هنگام استفاده از سرور های لینوکسی یا ویندوزی، نیازی نیست سیستم عامل کاربران نیز لینوکس یا ویندوز باشد.

نتیجه گیری :

ویندوز یک سیستم عامل است که به صورت فراگیر در سراسر دنیا، هم برای استفاده شخصی و هم در اکثر مراکز اداری و شرکت‌های دولتی و خصوصی در حال استفاده است و به دلیل جهان شمول بودن این سیستم عامل، اکثر شرکت‌های نرم‌افزاری دنیا، برای فروش نرم‌افزارهای خود، آن را تحت سیستم عامل ویندوز طراحی و تولید می‌نمایند که این امر نیز سبب تثبیت هرچه بیشتر این سیستم عامل در سطح دنیا شده است.

مراکز اداری و کارمندان سال هاست که از سرورهای ویندوزی در ادارات استفاده می‌نمایند و به آن عادت کرده اند و انتقال سیستم عامل ویندوز به یکی از توزیع های گنو/لینوکس هزینه بالایی برای مراکز اداری خواهد داشت. هم هزینه مالی و هم هزینه زمانی، زیرا باید برای کارمندان دوره‌های آموزشی کار با سیستم عامل لینوکس و نرم‌افزارهای آن گذاشته شود که ممکن است مورد توجه مدیران سازمانی واقع نشود، ولی اگر برای طولانی مدت، هم در زمینه سرورها و هم در زمینه دسکتاپ سرمایه گذاری شود، قطعا هم میزان بهره برداری چند برابر می‌شود و هم در هزینه‌ها صرفه جویی می‌شود. باتوجه به وضع کنونی و پیشرفت هرچه ببیشتر سیستم عامل گنو/لینوکس، برخی از شرکت‌های نرم‌افزاری تصمیم به تولید نسخه متن باز نرم‌افزار خود گرفته اند.

با نگاهی مهندسی به این موضوع به این نتیجه می‌رسیم که در واحد فنی هر کدام از مراکز اداری خصوصی یا دولتی، با حضور افراد متخصص در این زمینه و با توجه به ویژگی‌های منحصربه فرد سرور های لینوکس که در بخش های قبل به آنها اشاره کردیم، استفاده از این سیستم عامل در بخش سرور بسیار مفید خواهد بود، زیرا میزان اطمینان و امنیت را در بخشهای سرور افزایش داده و باعث کاهش هزینه می‌گردد و این کاملا برطبق منافع شرکت ها خواهد بود و نگاهی به آمارهای منتشر شده نیز گواه مهاجرت از ویندوز به لینوکس در بخش سرور خواهد بود.



(@Geek_072)



غول کش



فرض کنید سیستم عاملی در اختیار دارید که تنها عده ای خاص از افراد به صورت محدود از آن استفاده می کنند. این سناریو معمولا به شکل سخت افزاری انحصاری به همراه یک سیستم عامل بهینه شده ی یونیکسی برای مصارف خاص شرکتها به کار گرفته می شود. حال فرض می کنیم قصد دارید سیستم عاملی برای بیش از صد میلیون کاربر طراحی کنید، در قدم اول شما باید امکاناتی کاربر پسند برای جذب مخاطب خود فراهم کنید تا آنها دلیلی برای استفاده از سیستم عامل شما داشته باشند، شاید چندین سال پیش لینوکس یک ابزار در دست عده ای خاص برای مصارفی خاص بود اما به لطف کمپانی کنونیکال این دید کاملا در حال گذر است و ما در حال فراگیر شدن فرهنگ و دانش متن باز در بین افرادی با دانش و دید متفاوت هستیم. این شرکت انگلیسی که دفتر مرکزی اش در لندن قرار دارد در حال حاضر با ارزشی برابر پانصد میلیون دلار و رشد سهامی مناسب یک پشتوانه ی عالی برای توزیع محبوب ابونتو می باشد. با این تفاسیر کمی به بررسی شرایط این شرکت در بازار می پردازیم، این شرکت با داشتن چندین محصول در حال تجربه در زمینه های کاری مختلف می باشد مهم ترین محصول این شرکت همانطور که اشاره شد توزیع ابونتو است که در رقابتی نزدیک با مینت در رتبه دوم پرکاربرد ترین توزیع های متن باز قرار گرفته است اما امید این شرکت به بازگرداندن، عنوان محبوب ترین سیستم عامل متن باز به ابونتو دور از انتظار نخواهد بود.

هدف ابونتو و کنونیکال راحت تر کردن کار برای افراد خبره و تازه کار در یک قالب کاربری جالب به نام یونیتی است.



با توجه به آمار مناسب دانلود ابونتو می توانیم به کنونیکال نمره ی قبولی در رسیدن به اهدافش را اختصاص دهیم. کنونیکال علاوه بر تمرکز بر روی دسکتاپ دارای دیدی رو به جلو در رایانش ابری بر روی بستر **Open Stack** نیز هست.



به صورتیکه در حال حاضر ۶۰٪ سیستم عاملها بر روی راهکار ابری **Open Stack** ابونتو می باشند. از دست آوردهای دیگر کنونیکال می توان از ورود به بازار پر تشنج تلفن همراه اشاره کرد. ابونتو فون محصول متفاوت این شرکت در حال حاضر در حال مرحله ورود به عنوان یک گزینه ی متن باز برای کسانی است که به امنیت خود می اندیشند.

رمز موفقیت کنونیکال به صورت کلی از توضیحات قبلی را می توان در سادگی و راحتی برای کاربر خلاصه کرد جایی که کنونیکال بسیار موفق بوده است. کنونیکال علاوه بر تمام موارد گفته شد یک لطف بزرگ در حق جامعه ی متن باز کرده است و آن هم چیزی نیست جز راضی کردن استیم به سرمایه گذاری بر روی پلتفرم های لینوکسی که باعث خوشحالی بسیار زیاد در جامعه ی گیمر های لینوکسی شد، کسانی که همواره یک سیستم ویندوزی برای اجرای بازی مورد علاقه شان نیاز داشتند اکنون تنها نیاز به استیم دارند.

پس از این مقدار توضیح از نکات مثبت کنونیکال اشاره به نکات منفی این شرکت و انتقادات وارده بر آنها هم خالی از لطف نیست وقتی یک کمپانی شروع به قرار داد بستن و شراکت با کمپانی های بزرگی مثل **Cisco** و یا **Big Switch** می کند همواره شایعات و انتقاداتی در مورد رعایت حقوق شخصی افراد در محرمانه ماندن اطلاعاتشان مطرح می شود.



یکی از مهم ترین مسائل جنجالی کنونیکال با کاربران، سرویس ابونتو وان این شرکت بود که طی قرار دادی به آمازون پس از اسکن کردن آهنگها و یا فیلمهای کاربر گزینه های مناسب موجود برای خرید را به وی پیشنهاد می داد. این ویژگی هم اکنون در حالت پیش فرض غیر فعال شده است. کنونیکال همین طور یک سری دعاوی حقوق بر سر استفاده از نرم افزارهایش در سایر توزیع ها و همینطور با مینت بر سر یک سری تشابهات ساختاری داشت که به مرور زمان بر طرف شد و این شرکت این مشکلات را پشت سر گذاشت. در نهایت به نظر بنده کنونیکال یک نمونه ی شرکت موفق در جامعه ی متن باز می باشد که قصد دارد علاوه بر سود خود، به جامعه ی متن باز هم کمک کند و این رابطه تعریف کننده ی رابطه ی زیبای جامعه ی کاربری با شرکتها است.

(@AAP1024)

خاندان بزرگان



شاید برای خیلی از شما، این سوال پیش آمده باشد که چرا **Gnu/linux** ؟ در این مقاله ابتدا به بررسی هر یک از این موارد می پردازیم و درنهایت به جواب اصلی میرسیم.

اول از **GNU** شروع میکنیم تا به درک کامل چرایی استفاده از آن و چگونگی ایجاد آن نیز بپردازیم.

GNU در واقع یکسری ابزار و برنامه های رایگان است ، پروژه **GNU** در سال ۱۹۸۳توسط **Richard Stallman** به صورت عمومی در دسترس قرار گرفت و این دسترسی تا هم اکنون ادامه دارد.

GNU مجموعه ای از برنامه های کاربردی ، کتابخانه ها، ابزار توسعه و حتی بازی ها است.

حالا به توضیح خلاصه ای راجع به لینوکس میپردازیم :

حتما بیشتر شما حداقلِ اطلاعات را در مورد لینوکس دارید، ما نیز در اینجا به ذکر توضیح کوچکی راجع به این موضوع اکتفا میکنیم.

لینوکس یک سیستم عامل آزاد و متن باز است که کد منبع آن در اختیار همگان قرار دارد و همه می‌توانند در کدهای آن تغییر ایجاد کرده و بنا به نیازشان طراحی و استفاده کنند. آزاد و در دسترس بودن کدهای لینوکس سبب می‌شود تا بتوانید از طرز کارکرد دقیق سیستم‌عامل مطلع شوید.

بهتر است به این موضوع بپردازیم که چرا استفاده از **GNU/LINUX** یکی از بهترین روشهایی است که پیش رو داریم :

وقتی که عبارت **GNU/LINUX** را بکار میبریم در واقع لینوکس که همان سیستم عامل است در نقش هسته قرار خواهد گرفت، اما این کرنل(هسته) بدون برنامه های کاربردی **GNU** قدرت خاصی ندارد. پس به همین دلیل ترجیح بر این است که در کنار **LINUX** از پروژه **GNU** هم استفاده کنیم ، تا این دو در کنار هم یک مجموعه کامل را تشکیل بدهند. اکثر افراد به جای استفاده از **GNU/LINUX** از **LINUX** به تنهایی یاد میکنند که با توجه تجربیات بنده در این زمینه خیلی کارساز نخواهد بود .

چرا !؟ زیرا شبیه به این است که شخصی کاربر ویندوز باشد و روی ویندوز **Browser** ، **Media player** و حتی **Note pad** و .... نداشته باشد دراینصورت این ویندوز برای چه موردی میتواند استفاده شود و چقدر مشکلات و کارهای مورد نیاز کاربر را پیش خواهد برد؟! این دقیقا شبیه **linux** بدون **GNU** میباشد، اما با **GNU** میتوان به اضافه شدن خیلی از برنامه های لینوکسی مثل **GIMP** ، **Pidgin** ، **VLC** و **libereOffice** برای یک سیستم عامل خام (لینوکس) اشاره کرد.

گنو یک طرز تفکر بر مبنای آزادی اطلاعات است. برنامه هایی که زیر قرارداد **GNU** ارایه میشوند در واقع به کاربران خود این امکان را میدهند که بتوانند در کدهای آن برنامه هرگونه تغییری بر اساس نیازشان ایجاد کنند.

مجموعه برنامه هایی که در حال حاضر زیر قرارداد **GNU** ارایه میشوند شامل طیف وسیعی از برنامه های کاربردی میباشد که امکان نصب آنها روی هسته سیستم عامل لینوکس وجود دارد.

(@Master_0098)



قدرت یک هسته

همه سیستم عامل ها دارای یک هسته مرکزی هستند که مسئولیت آن، مدیریت درخواست هایی است که از بخش نرم افزار به بخش سخت افزار و یا بالعکس ارسال میشوند. درواقع اینگونه میتوان بیان کرد که کرنل از نظر ساختاری بین نرم افزار و سخت افزار قرار دارد. درخواست از طرف نرم افزار ارسال میشود و در هسته مورد بررسی قرار میگیرد تا نرم افزار بتواند از سخت افزار استفاده کند. همچنین کرنل پایین ترین سطح دسترسی را برای یک نرم افزار ایجاد میکند.



کرنل لینوکس در سال **1991** بوسیله یک شهروند فنلاندی به نام لینوس توروالدز که دانشجوی رشته علوم کامپیوتر بود طراحی و پایه گذاری شد. کرنل درهمه سیستم عامل های لینوکسی با توزیع های مختلف یکسان است و همه این سیستم عامل ها از یک هسته واحد استفاه میکنند. ولی هر یک از آنها با نرم افزارهای مختلف به این کرنل بنابر نیازشان شکل و طرح میبخشند.

اولین نسخه کرنل لینوکس در تاریخ **14** مارچ **1994** ارائه شد که در آن از یک پردازشگر با پایه **i386** استفاده می شد. به مرور مشکلات کرنل مرتفع گردید و بروزرسانی و اضافه کردن قابلیت های جدید به کرنل انجام شد. آخرین نسخه کرنل **4.4** است که در تاریخ **10** ژانویه **2016** ارائه شد.

کرنل از استفاده های چندگانه و بطور همزمان پیشگیری میکند. یعنی ممکن است گاهی بر حسب نیاز برنامه های در حال کار را متوقف کند تا این اطمینان حاصل شود که برنامه هایی با اولویت بالاتر بتوانند از پردازنده استفاده کنند. بطور مثال اگر برنامه ای در حال اجرا و همچنین در انتظار دریافت برخی داده ها نیز باشد کرنل آنرا در وضعیت انتظار قرار میدهد تا فضای استفاده از **CPU** آزاد شده و برنامه بتواند از آن استفاده کند.

دلایل بسیاری وجود دارد که کاربران بخواهند کرنل مورد نظر خودشان را پیاده سازی کنند، این بدین معناست که شاید بسیاری از کاربران بخواهند کد های مورد نیاز خود را در سیستم هایشان اجرا کنند و با این کار میتوانند از هدر رفتن وقت و حافظه جلوگیری نمایند، یکی از ویژگی هایی که کرنل را محبوب کرده قابل حمل بودن آن است. درواقع کرنل این قابلیت را دارد که در انواع مختلف سیستم ها و پردازشگرها به خوبی کار کند انواع مختلفی از پردازشگرها ازکرنل پشتیبانی میکنند که تعدادی از آنها را در اینجا نام میبریم:

Alpha, AMD, ARM, C6X, Intel, X86, Micro blaze, MIPS, PowerPC, SPARC, Ultra SPA

کرنل شامل گروه هایی است که به ۴ دسته طبقه بندی شده اند، در اینجا به تعریف هر یک از آن‌ها می پردازیم.

Monolithic Kernel:

نوعی معماری سیستم عامل است که سرتاسر سیستم عامل (درایو های سخت افزار، فایل سیستم ها و نرم افزار) در حالت سوپروایزر در فضای کرنل درحال کار کردن می باشد. این معماری قابلیت بارگذاری به صورت پویا را دارد و همچنین می تواند ماژول ها اجرایی را در زمان فعالیت خارج کند.

Microkernel:

در علوم کامپوتر مقدار حداقل نزدیکی نرم افزار را می گویند که می تواند مکانیزم های مورد نیاز برای پیاده سازی درخواست های کاربر و اعمال آنها روی سخت افزار را، از طریق کنترل سخت افزار ارائه دهد. این مکانیزم ها عبارتند از: مدیریت فضای آدرس پردازنده (**IPC**) و مدیریت فرایند پردازش را بر عهده دارد.

سیستم عامل هایی برپایه میکروکرنل هستند به شرح زیر می باشند:

AI, BeOS, Hurd, Mach, Mac OS X, MINIX و **QNX**

Hybrid Kernel:

یک نوع دیگر از معماری سیستم عامل است که سعی بر ادغام مزایا و موجودیت های **Microkernel** و **Monolithic Kernel** را دارد تا بتواند یک سیستم عامل مدرن پدید آورد، ساختاری که پشت کرنل **Hybrid** میباشد همانند **Microkernel** است اما برای پیاده سازی ساختار از روش **Monolithic** استفاده می شود.

سیستم عامل هایی که از این نوع کرنل استفاده میکنند عبارتنداز:

2000, Microsoft Windows NT, XP and new windows , Dragon Fly ,BSD

Exokernel :

یک نوع سیستم عامل می باشد که بوسیله **MIT Parallel** و گروه توزیع سیستم عامل توسعه یافته است، سیستم عاملی با توانایی برقراری قرارداد بین نرم افزار و سخت افزار، همچنین معماری **Exokernel** برای محافظت از منابع جداگانه به منظور تسهیل، مدیریت و سفارشی ساختن نرم افزارهای ویژه استفاده می شود و در سیستم عامل های ویندوزی و لینوکسی قابل اجرا است.

با توجه به مطالب فوق **kernel** عضو مهم سیستم عامل میباشد و مسولیت های فراوانی را پشتیبانی میکند.

(@Badrinex)

استاد بزرگ

در دنیای لینوکس بصورت کلاسیک وقتی نیاز به یک برنامه خاص داشته باشیم باید سورس کد آن را از جایی تهیه کنیم و سپس آن را بر روی لینوکس خود کامپایل کنیم تا بتوانیم از آن استفاده نماییم.

این موضوع با پیشرفت و رشد لینوکس و استفاده هر چه بیشتر از کتابخانه‌های مختلف متن باز و وابستگی هر چه بیشتر سورس کدها به منابع آن کتابخانه‌ها در قالب Dependency های مختلف کم کم باعث سختی و پیچیدگی خاصی شد که نیاز به یک برنامه برای مدیریت پکیجها احساس شد و کم کم برنامه‌هایی به اسم Package manager بوجود آمدند که مسئولیت دانلود کردن، نصب کردن، تنظیم کردن و پاک کردن پکیج ها را بر عهده گرفتند.

در حال حاضر هر یک از توزیع‌های معتبر دارای Package manager مخصوص خود میباشند که بصورت پیشفرض بر روی مخازن آن توزیع خاص تنظیم هستند تا آخرین پکیجهای موجود در مخازن را در اختیار کاربران خود قرار دهند.

در توزیع آرچ لینوکس Package manager معروفی به نام pacman وجود دارد که یکی از قدرتمندترین و کاملترین Package manager های موجود محسوب میشود.



Pacman را میتوان به نوعی قلب تپنده آرچ لینوکس دانست و به نوعی میتوان قدرت آرچ را مدیون وجود pacman در درون آن دانست.

Pacman برای استفاده در ترمینال ساخته شده است و به دلیل سادگی در برنامه نویسی خود دارای سرعت بسیار بالایی میباشد.

این Package manager از یک دیتابیس بصورت یک فایل متنی استفاده میکند که در واقع پکیج های سیستم را در آن نگهداری میکند و به کاربر اجازه میدهد که در هر بار یک یا چند یا همه پکیجهای خود را پاک، نصب، دانلود، آپگرید و یا تنظیم کند.

pacman از2002-02-25 که اولین ورژن خود را منتشر کرده است تقریباً بطور متوسط هر ۳ ماه یک بار یک آپگرید ورژن داشته است که این نشان دهنده قدرت جامعه قدرتمند برنامه نویسان پشتیبان آن است.

در حال حاضر آخرین ورژن pacman 4.2.1 است که در تاریخ 2015-02-20 منتشر شده است.

acman درواقع از چندین پکیج مختلف تشکیل شده است که هر یک مسئولیت کار خاصی را بر عهده دارند از مهمترین آن‌ها میتوان به makepkg, pactree, vercmp, checkupdates اشاره کرد.

در آرچ لینوکس pacman بطور پیش‌فرض از مخازن آرچ پکیجهای خود را دانلود و آپدیت میکند اما این امکان در پکمن وجود دارد که به راحتی و با اضافه کردن مخازن دیگر پکیجهای خود را از آن‌ها دانلود و نصب نمائید.

بطور مثال مخزن black arch به راحتی و با اضافه کردن آدرس آن به pacman قابلیت نصب مستقیم بر روی آرچ را بدست می آورد.

دستورات pacman بسیار ساده‌تر از دیگر Package manager ها مانند apt است.



قدرت واقعی آرچ لینوکس در استفاده از دو Package manager مختلف در کنار هم میباشد. در کنار pacman که مسئولیت مدیریت پکیجهای رسمی و مخازن معتبر در توزیع آرچ را به عهده دارد آرچ از برنامه دیگری به نام yaourt برای مدیریت پکیجهای غیر رسمی که در مخازن AUR آرچ موجود میباشد استفاده می کند و به این صورت قابلیت غلطان بودن آپدیت آرچ بهتر و پویا تر مدیریت میشود.

Pacman بخاطر تمرکز شدید بر روی ترمینال معمولاً گزینه اول کاربران عادی سیستمهای لینوکس محسوب نمیشود اما تقریباً اکثریت کاربران حرفه‌ای لینوکس بعد از امتحان بقیه Package manager حتی برای مدتی هم که شده از pacman بعنوان package manager خود استفاده میکنند.

(@BoBzBoBz)

Package Manager
Grandmaster
Linux Power



گیک و تلویزیون

Star wars



برای همه ی ما یک عنوان پر خاطره است حتی بسیاری از پدرها و مادر هایمان نیز با این مجموعه ی تاریخی خاطره دارند حال می خواهم به بررسی سری هفتم این مجموعه یعنی **The Force Awakens** بپردازم در ابتدا به این موضوع اشاره کنم که مجموعه ی جنگ ستارگان برای گیک ها دارای اهمیت خاصی می باشد شاید به دلیل همزمانی به راه افتادن فرهنگ و تفکر گیکی و یا داستان فوق العاده ی این مجموعه باشد که همواره جنگ ستارگان جذب کننده ی گیک ها بوده و هست. آخرین مجموعه ی این سری که اخیرا به کارگردانی **J.J.Abrams** به روی پرده های نقره ای رفت روایت قسمت اول سه گانه ی مجموعه ی بازگشت به جدال است. حال فرض می کنیم شما نمی دانید داستان جنگ ستارگان چیست و می خواهید از قسمت هفتم به تماشای این مجموعه بنشینید. شاید بسیاری از منتقدین بگویند که شما چیزی را از دست نداده اید قسمت هفتم بر خلاف موفقیت گسترده اش در گیشه به نظر بسیاری از منتقدین و طرفداران، داستان جدیدی نداشت و تکراری ازسری های قبلی با رنگ و لعاب جلوه های ویژه جدیدتر بود. هدف کارگردان با توجه به نظر منتقدین راضی نگه داشتن مخاطب پس از سالها غیبت این مجموعه، اقدام کارگردان جالب توجه به نظر میرسد. در هر حال اگر گیک هستید و یا دوستدار داستانهای علمی تخیلی و یا میخواهید یک فیلم جالب ببینید دیدن این فیلم احتمالا گزینه ی مناسبی برای شما است.

Scorpion



یک سریال پر طرفدار دیگر از شبکه ی **CBS** است که قصد بررسی آن را داریم. این شبکه در زمینه ساختن سریالهای علمی و تخیلی و به صورت کلی گیک پسند بسیار فعال است در گذشته به بررسی سریالهای **TBBT** و **CSI Cyber** از این شبکه پرداخته بودیم. حال قصد بررسی سریال **Scorpion** را داریم ژانر سریال همواره کمک زیادی به درک حال و هوای کلی یک سریال می کند. این مجموعه یک اکشن درام براساس زندگی نابغه ی معروف کامپیوتر **Walter O' Brien** است.از معروفترین اتفاق منصوب به وی می توان به هک ناسا در ۱۳ سالگی توسط وی اشاره کرد. والتر دارای هوش اجتماعی ۱۹۷ می باشد که بالاترین رکورد ثبت شده در طول تاریخ است. سریال با اشاره به این گذشته آغاز می شود و قصد نشان دادن گروهی به رهبری وی را دارد که قصد رفع مشکلات بزرگ از قبیل سقوط هواپیما و یا سایر داستانهای جانبی که در مسیر قهرمانان داستان رخ می دهد را دارد. فضای داستان دارای استرس،هیجان و ریتم تند است از این لحاظ می توان از آن به عنوان یک سریال سرگرم کننده برای گیک ها اشاره کرد اما شاید همین نکته در طول سریال به اغراق تبدیل می شود که برای بینندگان غیر منطقی به نظر برسد. به صورت کلی سریال و روابط عاشقانه ی والتر در طول مسیر و همینطور آشنایی با پیش خدمت رستوران و فرزندش حس خوبی به مخاطب منتقل می کند و از نقاط مثبت مجموعه به شمار می رود. در نهایت اگر به دنبال گذراندن وقت با اکشن و سرگرمی هستید این سریال قطعا انتخاب مناسبی است.

SiliconValley



من به شخصه سریالهای زیادی را دنبال می کنم اما ژانر مورد علاقه ام کمدی درام است پارادوکسی سخت و دوست داشتنی برای مخاطب و اعضای سریال است . سریال **Silicon Valley** یکی از سریالهای فوق العاده **HBO** است در ژانر ذکر شده است که در سایه ی سریال مطرح بازی تاج و تخت معمولا فضایی برای عرض اندام نداشته اما به نظر من در ژانر کمدی درام از بهترینها است. سریال در مورد بررسی و انتقادی طنز گرایانه به شرکتهای نوپا در حوزه تکنولوژی و همین طور چالش های پیش روی این شرکت ها در مسیر ورود به بازار است. شخصیت اول داستان مانند کلیشه ی همه داستانهای کامپیوتری فردی باهوش، خوش فکر و گوشه گیر است اعتماد به نفس پایین در وی موج می زند که همین موضوع توانایی بیان درست ایده هایش را تا حدی از وی گرفته است. سریال از نظر من با موضوع جدی و شوخی های بسیار ساده هوشمندانه ش طنزی بسیار سطح بالا را با خود یدک می کشد. من به شخصه از سریال لذت فراوان بردم رضایت بینندگان سریال تا حدی بوده که مدیر عامل بنیاد **Mozilla** یک نمایشنامه کوتاه برای یک قسمت این سریال نوشته و برای نویسندگان به عنوان نتیجه گیری نهایی اینکه اگر گیک هستید، این سریال را از دست ندهید.

(**@AAP1024**)

گوهر آدمیت

اجازه بدهید علت محبوبیت اوبونتو را با معنی لغت آن که ریشه در داستان زیر دارد بیان کنم. در آفریقا مسابقه ای بین بچه ها برگزار شد. سبدی از میوه در نزدیکی درختی گذاشتند و قرار بر این شد هر فرد زودتر به آن سبد برسد میوه های خوشمزه را از آن خود کند. با شروع بازی تصویر خیره کننده ای به چشم خورد و آن اینکه تمامی بچه ها دستان یکدیگر را گرفتند و با هم به سمت سبد دویدند و همگی شروع به خوردن میوه های خوشمزه کردند. پس از آن از تک تک بچه ها سوال شد: در حالی که یک نفر از شما می توانست به تنهایی همه میوه ها را بخورد چرا از هم جلو نزدید؟ آن ها گفتند: چگونه یکی از ما می تواند خوشحال باشد در حالی که دیگران ناراحتند؟!!



اوبونتو نیز بدین معناست که "آدمی به کمک انسانهای دیگر به گوهر آدمیت می رسد" و فلسفه محبوبیت نرم افزار های متن باز نیز از این لغت مشتق شده است. نسخه های متخلف به صورت رایگان در دسترس است، هر کدام از شما می توانید با توجه به نیاز خود آن را بهینه سازی و در اختیار همگان قرار دهید و به قانون بنیاد نرم افزار های رایگان پایبند باشید.

اوبونتو سادگی و سرعت بالا و کاربر پسند بودن در رده بالایی قرار دارد، که از دو طریق تجارت و کاربران لینوکس در سراسردنیا در حال توسعه می باشد.

یکی از ویژگی های اصلی پروژه **Ubuntu** تاکید بر دسترسی همگانی و جهانی سازی آن است، بدین علت اوبونتو در اکثر زبانهای زنده دنیا پشتیبانی می شود.



اوبونتو در نسخه های مختلف هسته اصلی خود را برای کاربران **desktop** عرضه کرده است. به ادعای آقای شاتل ورث موسس شرکت کنونیکال **LTD** پشتیبان **Ubuntu** :"مایکروسافت نقش عمده ای در بازار کامپیوترهای **Desktop** دارد و این یک باگ است که اوبونتو آمده است تا این باگ را تعمیر کند".

از نقاط قوت **Ubuntu** می توان به حذف برنامه های اضافی و استفاده از حداقل سخت افزار جهت نصب اعم از ۲۵۶مگا بایت حافظه رم و پردازنده ۱,۷ سلرون و هارد ۵ گیگا بایت اشاره کرد و هم چنین مراحل نصب آن نیز بسیار آسان می باشد. اوبونتو را میتوان به صورت **Live** نیز تست کرد، بدین شکل که با قرار دادن **CD** در **CD-Rom** و بدون نیاز به نصب آن را اجرا کرد.

Ubuntu نیز مانند سایر نسخه های لینوکس هر ۶ ماه یک بار منتشر می شود و هر ۲ سال یکبار نسخه **Long time support** به معنی پشتیبانی بلند مدت آن انتشار می یابد.

از قابلیت های **Ubuntu** می توان به تنظیمات میزکار **Unity** در اوبونتو که بسیار ساده و لذت بخش است نام برد و هم چنین نرم افزار **Libre Office** معادل نرم افزار **Microsoft Office** ویندوز می باشد که به صورت پیش فرض در **Ubuntu** نصب می باشد.

نا گفته نماند که امنیت بالا و سطح دسترسی به جهت جلوگیری از حذف های بی مورد در لینوکس از درجه اهمیت بالایی برخوردار است.

برای کاربران تازه وارد مدیریت بسته های گرافیکی توسط برنامه **Gnome PAckageKit** که به طور پیش فرض نصب است، میباشد که روشی برای نصب آسان نرم افزارهاست.

و همچنین برای نصب برنامه های که به طور پیش فرض نصب نیستند می توانید به مرکز نصب نرم افزار اوبونتو یا **Ubuntu Software Center** رفته و نرم افزارهای مورد نیازتان را نصب نمایید.

در آخر اینکه قابلیت های اوبونتو آن چنان زیاد است که اگر کم حوصله نیستید و هیجان یادگیری متفاوت را دارید، تنها کافی است **Ubuntu** را نصب کنید و از محیط جذاب آن لذت ببرید و از کاربران همیشگی آن شوید.

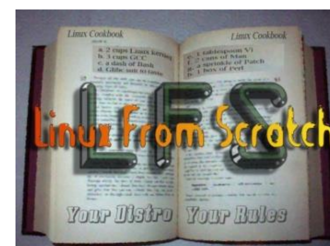


(@mehnaty)



هفت خان

آیا تا به حال به این فکر افتاده اید که برای خود یک توزیع لینوکس بسازید؟ اگر با توزیع های لینوکس سروکار داشته باشید به احتمال زیاد حداقل یکبار به این فکر افتاده اید. ایده‌ی اصلی پروژه ی لینوکس از پایه یا همان **linux from scratch**(**lfs**) این است که هر کس بتواند برای خود یک توزیع اختصاصی لینوکس بسازد و راهی برای برپایی سیستم عامل لینوکس بوسیله ساختن همه اجزا به صورت مرحله به مرحله است. هر مرحله با استفاده از ابزار های **command line** کامل می شود، که این روند نیاز به یک سیستم عامل لینوکس از پیش، نصب و راه اندازی شده دارد که حداقل موارد مورد نیاز کامپایل شدن **lfs** را داشته باشد.



ساختن **lfs** را می توان با ساختمان سازی مقایسه کرد. **Lfs** اسکلت ساختمان را به ما می دهد اما بستگی به تصمیم خودمان دارد که آیا برای این بنا نیاز به لوله‌کشی و سیم‌کشی و آشپزخانه و سرویس بهداشتی و... داریم یا نه، تا بتوانیم این ساختمان را به هر مکانی مثل خانه، مطب، مدرسه یا ... تبدیل کنیم. به همین منوال ممکن است سیستم ما برای دسکتاپ و سرور و روتر و یا استفاده های اختصاصی دیگر باشد مانند اکثر سیستم عامل‌های دیگر، توزیع لینوکسی جدید معمولاً روی یک پارتیشن تخصیص یافته نصب می شود.

روش پیشنهادی استفاده از یک پارتیشن خالی موجود، یا در صورت داشتن فضای کافی ساخت آن است. یک سامانه‌ی کمینه نیاز به پارتیشنی حدود ۸/۲ گیگابایت (**GB**) دارد که برای ذخیره‌ی همه‌ی تربال های منبع و کامپایل بسته ها کافی است.

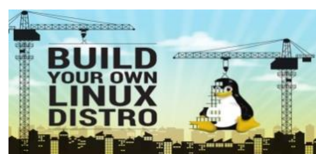
به هرحال اگر سیستم عامل جدید قرار است به عنوان سامانه‌ی گنو/لینوکس ابتدایی استفاده شود، احتمالا باید یک نرم افزار اضافی نصب شود که نیاز به فضای بیشتری دارد. یک پارتیشن ۱۰گیگابایتی فضای معقولی است. باید توجه داشت که خود سیستم عامل جدید فضای نسبتا کمی را به خود اختصاص میدهد. کامپایل کردن بسته‌ها می‌تواند نیاز به مقدار زیادی فضای دیسک داشته باشد که بعد از نصب بسته خالی می شود.

پیاده‌سازی یک توزیع لینوکسی جدید نیاز به دانستن اصول طراحی نرم‌افزار و برنامه‌نویسی به زبان‌های سی، سی پلاس پلاس، جاوا، پایتون، اسمبلی و ... دارد و باید کامپایل کردن و تنظیمات مختلف سیستم مثل **grub, xorg, ...** را بدانیم.

(که تنظیمات و کامپایل کرنل را نیز شامل می شود) و در آخر باید یک دیسک زنده از آن ساخته شود.

مزایای داشتن یک توزیع اختصاصی لینوکس انعطاف‌پذیری و امنیت است. هم‌چنین درک بیشتری از فعالیت‌های داخلی یک سیستم عامل مبتنی بر لینوکس به ما می‌دهد. هنگامی که از یک توزیع خاص لینوکس استفاده می کنیم (مثلا**redhat**) یک سری برنامه برایمان نصب می شود که هیچ نیازی به آنها نداریم و هیچوقت قرار نیست از آنها استفاده کنیم که باعث اشغال شدن بخش زیادی از هارد دیسک می شود. می توان یک توزیع لینوکسی (**lfs**) با حجم کمتر از ۱۰۰ **mb** نوشت یا حتی خیلی کمتر مانند یک توزیع خاص که فقط **Apache web server** را اجرا کند و حجم آن فقط **mb** ۸ باشد!

مهم تر از همه موضوعی که همیشه و همه جا اهمیت دارد، داشتن امنیت بیشتراست.



وقتی یک توزیع برای خودمان بسازیم توانایی رفع مشکلات امنیتی سیستم خود را داریم و نیازی نیست منتظر پکیج های باینری باشیم، تا بتوان به وسیله آنها چاله های امنیتی سیستم را بر طرف کرد.

همانطور که متوجه شده اید لینوکس از پایه پروژه ی زمانبری است و علاوه بر مطالعه فراوان، حفظ انگیزه در طی مراحل انجام فرآیند ضروری است اما با استفاده از روحیه گیکی مطمئنا امکانپذیر است.

(@Emitiss)



مصاحبه با یک گیک

در این قسمت گفتگویی خواهیم داشت با آقای آرکوش (Arcush@) یکی از مدیران ویکی رسمی فارسی زبان آرچ لینوکس و با ایشان و همکارانشون بیشتر آشنا خواهیم شد.

بابز : برای شروع لطفا خودتون رو برای خوانندگان قبیله گیک ها معرفی کنید و کمی در مورد سوابق گیکی خودتون برای ما بگید تا بیشتر با شما آشنا بشیم.

آرکوش: اسم مستعار من که در دنیایی مجازی به آن شناخته می‌شوم (آرکوش) است. من ۲۹ ساله هستم و ساکن (ایران) دانشجوی سال آخر دکترای ریاضیات هستم و قصد دارم رشته فیزیک با گرایش نسبیت را در دانشگاه ادامه دهم.

بصورت شخصی و بخاطر علاقه فراوان ادبیات فارسی را نیز مطالعه میکنم و همچنین در کلاسهای مربوط به آن شرکت میکنم. یکی از افتخاراتم این است که چند سال سر کلاسهای دکتر شفیعی کدکنی حاضر شدم.

رشته ورزشی که تا سال گذشته بصورت حرفه‌ای دنبال میکردم (کشتی) بود و در کنار آن کوهنوردی رو هم بعنوان ورزشی مفرح دنبال میکنم.

بابز: خوب چند وقت است که با کامپیوتر و دنیای مجازی آشنا شده‌اید و در این مدت چه کارهایی را انجام داده اید؟

آرکوش: من تقریباً حدود ۹ سال است که با کامپیوتر و دنیای آیتی در ارتباط هستم که از این مدت ۴ سال آن را با ویندوز و محصولات مایکروسافت مشغول بودم در این مدت از ویندوز XP تا ویندوز ۷ را تجربه کردم و بعد از آن به دنیای لینوکس مهاجرت کردم و هم‌اکنون حدود ۵ سال است که در دنیای لینوکس فعال هستم. در این مدت یک هفته از توزیعهای (فدورا) و (اوپن سوز) استفاده کردم و سپس به مدت ۴ سال از توزیع (ابونتو) و حالا حدود بیشتر از یک سال است که از توزیع (آرچ)استفاده میکنم.

بابز: بر روی کامپیوتر شخصی خودت از چه نرم افزارهایی تخصصی استفاده میکنی؟

آرکوش: حقیقتش اینه که من خیلی با ابزارهای کامپیوتری درگیر نیستم و بیشتر به کامپیوتر به دیده fun نگاه میکنم اما شاید تنها ابزار تخصصی که میتونید روی کامپیوتر من پیدا کنید نرم‌افزارهایی مشابه (متلب) است. من قبلا روی ویندوز از متلب و latex استفاده می کردم. اما روی لینوکس از sage, octave و latex استفاده می کنم.

بابز: چرا بین این همه توزیع مختلف لینوکس تصمیم گرفتی از (آرچ) استفاده کنی؟

آرکوش: من این انتخاب رو کاملاً آگاهانه انجام دادم. معتقد هستم که (ابونتو) در حال انجام حرکات اشتباهی هست که با تئوری و ایدئولوژی گنو مطابقت نداره و این موضوع دلیلی بود که من رو متقاعد کرد تا از (ابونتو) به (آرچ) مهاجرت کنم. در زمان انتخاب توزیع با شایعات بسیاری در مورد (آرچ) رو به رو شدم. اول اینکه گفته میشه (آرچ) ناپایداره . تجربه شخصی من میگه پایداری (آرچ) حداقل از (ابونتو) بیشتر است چیزی که اون رو ممکنه ناپایدار کنه استفاده غلط از اون است که ما سعی کردیم در انجمن رسمی فارسی زبان (آرچ) لینوکس کاربران را به استفاده صحیح و توأم با آگاهی از (آرچ) توصیه کنیم.

دومین شایعه این بود که (آرچ)بخاطر (غلطان) بودن باعث حروم شدن حجم اینترنت میشود. که باید در این مورد بگم که (آرچ) ویرایشهای متفاوت نداره بلکه شما بعنوان کاربر یک بار اون رو نصب میکنید و در مرور زمان قطعات کوچک آپدیت را به آن اضافه میکنید. ویرایش های lts برخی توزیع ها چیزی بجز انتظار برای کاربر به ارمغان نمیارن. انتظار اینکه چند سال بعد قراره چه تحولی اتفاق بیفته. البته بحث من روی سرور ها نیست. قضیه اون ها کاملا جداست. اما درحال حاضر برای من قابل قبول نیست که یک توزیع، بروز نباشه و انبوهی از بسته ها بی دلیل روش نصب باشن. البته این یک نظر کاملا شخصیه. یک چیزی هم که باید حتما بهش اشاره کنم اینه که ما توزیعی تحت عنوان توزیع حرفه ای نداریم. اما اشخاص حرفه ای هر تعریفی که داشته باشن، در یک شرط باید صدق کنن و اون شرط اینه که با هر توزیعی بتونن کنار بیان. اما هر شخص مختاره تا دیدگاه کاملا اختصاصی خودش رو داشته باشه. این کاملا جدای از حرفه ای گریه. بهرحال، من فکر می کنم توزیع مناسب من آرچه و تابحال هم خوب سواری داده. یک حقیقت محض هم وجود داره: توزیع های که از ابتدا شما رو درگیر باطن خودشون می کنن درسته که مقداری وقت از شما خواهند گرفت، اما چند برابر اون، دانش هم به شما خواهند بخشید. پس اگر آرچ رو تجربه نکردید، حتما امتحانش کنید. ضمناً (آرچ) کرنل استیبلی دارد که با نصب اون کرنل از core بسته های stable مخازن استفاده خواهد کرد و به نوعی میشه گفت دیگه حتی (توپ) هم نمیتونه تکونش بده. و البته این موضوع رو هم بد نیست بگم که اگر حال و حوصله نصب در محیط ترمینال رو ندارید میتونید از نسخه هایی مثل (آنترگوس) و یا (مانجارو)هم استفاده کنید

بابز: در آینده قصد نداری توزیع‌های دیگر لینوکس رو امتحان کنی؟

آرکوش: اتفاقاً بدلیل علاقه‌ای که به تفکرات (استالمن) دارم، خیلی دوست دارم ۲ توزیع (اسلکور) و (جنتو) رو امتحان کنم ولی متأسفانه هنوز وقت کافی برای این کار پیدا نکردم.

بابز؛ خوب کمی در مورد انجمن و ویکی فارسی توزیع آرچ لینوکس برامون توضیح بده.

آرکوش: آرچ یه انجمن راکد داشت که آرش (AraaaX@) سعی کرد با ارتباط با مدیرانش اون رو احیا کنه و بنابراین خود آرچ هم در ایران بیشتر شناخته بشه. اما متاسفانه موافقت نشد. ما تصمیم گرفتیم یه انجمن بسازیم و به موازاتش یه ویکی. الان توی ماه پنجمش هستیم و کارا خوب داره پیش میره. درواقع از انتظاراتی که داشتیم فراتره. ما البته تبلیغ زیادی هم نکردیم و حتی توی انجمن تالارهای کافه و آف تاپیک نزدیم. دلیلمون هم اینه یه انجمن خالص باشه و در کمال مهربانی از همدیگر چیز یاد بگیریم. الان توی موتورهای سرچ، تاپیک ها و لینک های انجمن دارن بالا میان و ازین نظر به هدف رسیدیم. راجع به ویکی متاسفانه تعداد افراد کم هستن. ما یه تاپیک

توی انجمن زدیم که هر کسی هر صفحه ای رو که مایل هست بدون هیچ ددلاینی ترجمه کنه. تاحالا بیش از 40 صفحه ترجمه شدن و پنج شیش تاشون هم درحال ترجمه هستن. نزدیک به ده صفحه هم آموزش های بومی داریم که درحال تکمیل شدن هستن. ویکی آرچ بدون اغراق، کاملترین منبع گنولینوکسه. این رو همه اذعان دارن. بنابراین کار ما خیلی سهمگینه و توقع نداریم یه شبه به هدف برسیم. بلکه برنامه ریزی ما حرکت روی یک شیب صعودی هست. ما مطمئن هستیم کمتر از دو سال کاملا پایدار میشیم و یک دیتابیس غنی هم در اختیار جامعه گنوی ایران قرار میدیم. فروم و ویکی آرچ ما بصورت رسمی نماینده فروم و ویکی اصلی آرچ هستند.

بابز: نظرتون در مورد جامعه متن باز ایران در حال حاضر چیه؟

آرکوش: بنظر من دو تا عامل باعث شدن تا جامعه لینوکس ایران درجا بزنه:

اولیش توهم همه چیز فهمی بین اعضای این جامعه هست. البته همواره منظورم از اعضا، اکثر اعضا هست. این توهم تا جایی پیش میره که اعضای این جامعه فکر می کنن با دیدن چهارتا کلیپ توی یوتیوب میتونن عمل جراحی هم انجام بدن. متاسفانه این یه تابو شده و به نظر من رفته رفته در حال بدتر شدنه. مخصوصا اونایی که درگیر انجمن ها میشن، ممکنه به این ویروس هم آلوده بشن.

دومیش برمیگرده به گم کردن هدف. به نظرم هیچ اشکالی نداره اگه کسی بیاد و پز توزیعش رو بده. اما تا جایی که بی احترامی انجام نگیره. همچنین این بحث های کدوم توزیع بهتره که در تمام دنیا وجود داره در صورتیکه با ارایه دلیل باشه باعث بالا رفتن فهم عمومی میشه. اما متاسفانه ما اینطور رفتار نمی کنیم و عمده کاری که بعضی ها انجام میدن، تخریب هست.

درکل فعلا این جامعه قیافه چندان جالبی نداره. از لحاظ علمی هم بدون تعارف باید بگم که تعطیله. اما باید به آینده امیدوار بود.

بابز: خیلی از اینکه وقتتون رو در اختیار ما قرار دادین از شما متشکریم و برای شما و دوستان دیگرتان در انجمن و ویکی آرچ لینوکس آرزوی موفقیت میکنیم.

(@BoBzBoBz)



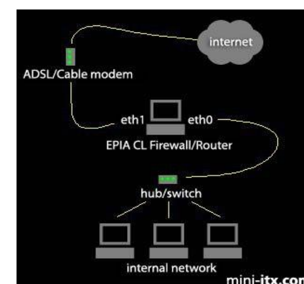
قدرت نرم

شبکه های کامپیوتری بسیار گسترده هستند و برای اتصال آنها به هم نیاز به مسیریابهایی در دل شبکه می باشد تا شبکه های مختلف را برای ارسال داده از رایانه ای به رایانه ای دیگر آماده کند، به طور معمول برای اینکار از روتر استفاده میشود.



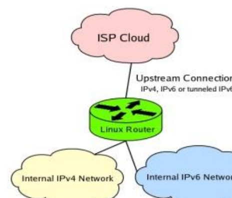
دستگاه های روتر مختلفی وجود دارند که از آن جمله می توان روتر سیسکو و میکروتیک را نام برد. اما در مواقعی که دستگاه روتر وجود ندارد چکار باید کرد؟ سیستم عامل های ویندوز و لینوکس این قابلیت را دارند و می توان از آنها به عنوان روتر استفاده کرد. در این متن ما در مورد اینکه چرا از لینوکس به عنوان یک روتر استفاده می کنیم، استفاده از روتر لینوکس چه مزیت و معایبی دارد و چه راه حل های وجود دارد برای اینکار، بحث خواهیم کرد.

چرا از لینوکس به عنوان یک روتر استفاده می کنیم ؟

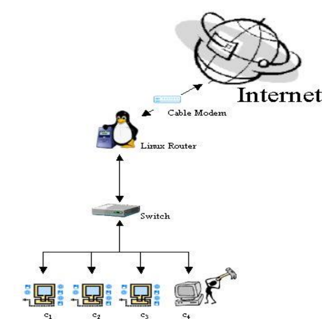


دلایل زیادی وجود دارد که می توان اقدام به انجام این کار کرد :

لینوکس دارای قابلیت های زیادی برای روتینگ می باشد و می توان از ابزار های **route, iptable** و **firewall** استفاده کرد و امنیت بیشتری را برای انجام عمل روتینگ اعمال کرد، همچنین اکثر دیوایس هایی که به عنوان یک روتر در شبکه فعالیت می کنند دارای سیستم عامل پایه لینوکسی هستند، اما شرکت های زیادی با تغییرات و طراحی که در آن بوجود می آورند آن را مختص به دیوایس های شرکت خود می سازند. اما ما چرا نباید از خود سیستم عامل لینوکس برای اینکار استفاده کنیم؟ هم به سادگی در دسترس است و همچنین نیاز نیست در یک شبکه، از دیوایس های جدا برای اینکار استفاده کنیم. یکی مزایایی که استفاده از لینوکس به عنوان یک روتر به ما میدهد در بر داشتن هزینه کمتری نسبت دیگر روش ها است، به عنوان مثال هزینه خرید یک روتر سیسکو بیشتر از **500$** میشود.



اما سیستم عامل لینوکس برای کاربران در دسترس است و فقط کافی است پکیج های مربوط به آن را نصب کنیم، مزیت دیگر که می توان مربوط به آن اشاره کرد این است که محدودیتی در رابطه با ارتقا حافظه و **cpu** سیستمی که قرار است به عنوان روتر لینوکسی باشد وجود ندارد، درصورتی که دیوایس های دیگر دارای محدودیت هایی هستند. مدیریت روتر لینوکسی به مراتب بهتر از دیوایس های دیگر است و دست کاربر برای کانفیگ بازتر می باشد. از معایب میتوان به این مورد اشاره کرد که اگر بر روی کیس یک کامپیوتر، روتر لینوکس را راه اندازی کنیم استیبل نمی باشد.



A Linux Router and some computers on a standard Ethernet LAN (that's mine on the right).

تعدادی از توزیع های مبتنی بر لینوکس که کار اصلی آنها روتینگ است و فقط کافی است آنها را بر روی سیستم خود یا ماشین مجازی نصب کنیم و از امکانات آن استفاده کنیم عبارتند از:

DD-WRT
RouterOS
Untangle
ZeroShell
Talisman Sveasoft
Vyatta
pfSense
CoovaAP
Tomato Firmware

و در آخر می خواهیم مقایسه کوچکی داشته باشیم که در روتر لینوکس و روتر سیسکو چه پروتکل های پشتیبانی می شوند و چه پروتکل های پشتیبانی نمی شوند. در روتر های سیسکو از پروتکل های **BGP,MP-BGP,eBGP/iBGP,OSPF,EIGRP,RIP** و **IS-IS** پشتیبانی می شود و در روتر لینوکسی همه روتینگ پروتکل ها پشتیبانی می شوند به جز پروتکل **EIGRP** که مختص خود شرکت سیسکو می باشد.

(**@Badrinex**)



انواع ذره بین

برای مانیتورینگ شبکه نرم افزار های متعددی وجود دارد که به تعدادی از این نرم افزارها در اینجا اشاره کرده و شرح مختصری راجع به هرکدام مینویسیم:

Nagios:

نگیوس یکی از نرم افزارهای پر طرفدار متن باز در حوزه ی مانیتورینگ سرویس های شبکه میباشد. این نرم افزار نمایی از سرویس ها و هاست ها و هشدارهایی در خصوص وضعیت سرویس ها به کاربران شبکه نشان می دهد.

نگیوس در ابتدا تحت اسم **Netsaint** ایجاد و نوشته شده بود، "**Sainthood**"(یا تقدیس) که مرجعی برای اسم اصلی این نرم افزار می باشد، در پاسخ به رقابت قانونی با مالکان مارک های تجاری مشابه، مجبور به تغییر شد. **Nagios** یک نرم افزار متن باز و نیز برنامه ای کاربردی برای نظارت شبکه می باشد. این نرم افزار در ابتدا به منظور کار تحت **GNU/Linux** طراحی شده بود، اما در حال حاضر روی یونیکس های گوناگون دیگر نیز به خوبی اجرا می شود.

در ادامه به بررسی ویژگی های این برنامه میپردازیم:

- مانیتورینگ سرویس های شبکه مانند **SMTP** **pop3,HTTp,NNTP,ICMP,SNMP,FTP,SSH**
- مانیتورینگ بر روی همه وضعیت های شبکه از قبیل مشکلات هاست ها، دما، هشدارها...
- کنترل سرور از را دور توسط سرویس های**SSH** یا **SSL**
- حتی می توانید توسط زبان های **Python , Ruby , php , c++ , shell** پلاگین های ساده ای بنوسید که اطلاعات بیشتری از وضعیت شبکه بدست آورید.
- پلاگین هایی برای ترسیم نمودار های داده ای و حتی یک امکان جالب برای ارسال پیام هنگام بروز مشکل یا برطرف شدن آن، برای هاست ها یا سرویس ها از طریق **Email** یا **SMS** و یا حتی پلاگین های موجود
- امکان بکاپ گرفتن از تمام فایل های **log**
- داشتن واسط گرافیکی زیبا

شما میتوانید با مراجعه به سایت **http://www.nagios.org** ازامکانات این نرم افزار قدرتمند استفاده کنید.

Zabbix:

یکی از نرم افزارهای رایگان مانیتورینگ شبکه Zabbix است . این نرم افزار برای مانیتور کردن داده ها نیاز به دیتابیس های MySQL،PostgreSQL، SQLlite، Oracle،IBM DB2 دارد.

زابیکس مانیتورینگ سرویس های شبکه، سرورها و سخت افزار شبکه را انجام می دهد، همچنین این نرم افزار از سمت سرور نیاز به زبان C و چون تحت وب است نیاز به php دارد. این نرم افزار قادر به چک کردن وضعیت چند سرویس ساده مانند **SMTP** یا **HTTP** بدون نصب هیچ برنامه ای از سمت کلاینت می باشد. همچنین زابیکس بر روی سیستم عامل های یونیکسی و ویندوزی نصب می شود و پارامترهایی نظیر میزان استفاده از **CPU**، ترافیک شبکه، میزان فضای استفاده از هارد دیسک و غیره را کنترل می کند. همچنین با انجام تغییراتی در هنگام نصب زابیکس سرویس هایی نظیر **SNMP** و**TCP** و **ICMP** و به همان خوبی سرویس های **IPMI**، **SSH**، **telnet** را بر روی کلاینت ها مانیتور میکند. این نرم افزار از مکانیزم اخطار برای سیستم های **real-time** مانند **XMPP** پشتیبانی می کند. **Zabbix** از سه ماژول مهم زیر پیروی می کند:

· Server (written in C)
· Agents (written in C)
· Frontend (PHP and Javascript)

Munin:

یکی دیگر از نرم افزارهای متن باز مانیتورینگ شبکه، یا سیستم ها می باشد. که خروجی را به صورت گرافیکی و تحت وب به کاربر شبکه نشان می دهد علاوه بر اینکه کار کردن با این نرم افزار بسیار راحت و آسان است، حدود ۵۰۰ پلاگین از پیش تعریف شده برای این نرم افزار وجود دارد که به قابلیت های این نرم افزار اضافه می کند. شما با استفاده از این نرم افزار به راحتی می توانید وضعیت های مختلف سیستم ها ، شبکه و **SANs** خود را مانیتور کنید.

در واقع این نرم افزار بیشتر به دنبال پیدا کردن تفاوت های شبکه در هر روز است و همچنین برای تهیه گزارش ها توسط پکیج **RRDtools** که به زبان **Perl** نوشته شده استفاده می کند. منطق این نرم افزار به این صورت است که شامل یک نود اصلی و چندین نود فرعی است، که نود اصلی توسط ارتباط برقرار کردن با سایر نود ها اطاعات مربوط به وضعیت شبکه را جمع آوری می کند و توسط پکیج **RRDtools** که بر روی سرور یا سیستم نصب شده است، به صورت نمودار یا گراف به کاربر شبکه نشان می دهد. از ویژگی های این نرم افزار می توان قابلیت تعریف پلاگین های جدید را نام برد.

MRTG:

یکی از قدیمی ترین و در عین حال ساده ترین نرم افزار های مانیتورینگ می باشد که بصورت گراف پارامتر های در حال نظارت را نمایش می دهد، اطلاعات خود را در فایل های **RRD** ذخیره کرده و از پایگاه داده ای استفاده نمی کند. این نرم افزار با استفاده از **Perl** نوشته شده و بر روی **Linux,Windows,Mac** قابل استفاده می باشد. وب سایت این نرم افزار شامل اسکریپت هایی است که برای نظارت بیشتر بر مواردی همچون **Sql,Firewall,Cpu,Ram** کاربرد دارد. این نرم افزار کاملا رایگان بوده و از وب سایت مربوطه قابل دانلود می باشد. نصب آن بسیار ساده بوده و امکانات خاصی را در اختیار شما قرار نمی دهد. برای مانیتور کردن هر نود نیازمند**OID** های مربوطه خواهید بود. هیچ تفکیک و یا گروه بندی نمی توانید برای نود ها ایجاد کنید، از اطلاعات آن باید بصورت دستی پشتیبان تهیه کرد و هیچ مکانیزم پشتیبان گیری وجود ندارد. برای

محیط های بزرگ توصیه نمی شود. هرگونه تغییر در برنامه را باید خودتان انجام دهید.

در این قسمت به بررسی نرم افزار MRTG خواهیم پرداخت :

امروزه MRTG به عنوان یکی از مهم ترین ابزار Monitoring برای کنترل ترافیک Link های مختلف درشبکه ها شناخته می شود.

MRTG براساس پروتکل SNMP پایه گذاری شده است که به صورت پیش فرض v1 SNMP و درصورت نیاز v2 SNMP مورد استفاده قرارمی گیرد. بااستفاده از MRTG می توان کلیه تجهیزات شبکه را که از پروتکل MRTG پشتیبانی می کنند, مانیتور نمود.MRTG با استفاده از SNMP اطلاعات مورد نیاز جهت Monitoring را دریافت وآنهارا به صورت Graph نمایش می دهد.

معمولاً مدیران شبکه با مشکلات فراوانی درشبکه مواجه میشوند، که بسیاری از این مشکلات ممکن است به صورت نهفته برای مدتها باقی بماند. برخی ازاین مشکلات به قدری جدی هستند که اگر به سرعت مرتفع نگردند خسارات مالی زیادی به همراه خواهند داشت .

لذا نیاز به یک نرم افزار Monitoring جهت کنترل ترافیک شبکه غیر قابل اجتناب است.

اکثر خرابی های شبکه به دو دسته عمده تقسیم می شود:

۱. down شدن Link یا Device ها

۲. افت کیفیت بازدهی شبکه

شماری از توانمندی های MRTG در برخورد با مشکلات مذکور عبارتند از :

۱. down شدن Link یا Device ها

درصورت قطع شدن Link یا ایجاد مشکل در Device ها با کنترل گرافهای MRTG ، به سرعت می توان به خرابی مورد نظر پی برد ، وآن رابرطرف نمود. این مورد به خصوص درشبکه هایی که تعداد Linkها وDevice های آن زیاد باشد و با توجه به این موضوع کنترل تک تک آنها وقت گیرخواهد بود، بسیار حائز اهمیت است، چراکه بامشاهده ی نمودارهای MRTG می توان به آسانی وضعیت کل شبکه را کنترل نمود.

تحلیل آماری :

باجمع آوری اطلاعات مربوط به Link ها در گرافهای روزانه ، هفتگی ، ماهانه و سالانه ، می توان به رفتار ترافیکی Link ها پی برد . که برخی ازآنها عبارتند از :

مدت زمان Peak : زمانهایی که بیشترین ترافیک از Link ها عبور می کند .

Peak Rate :بیشترین مقدار ترافیک دریک بازه زمانی .

Average Rate : مقدار متوسط مصرف .

این اطلاعات می تواند برای مدیر شبکه در تصمیم گیری های ارتقاء و پیشبرد شبکه بسیار کارساز باشد.

پیداکردن مشکلات نهفته :

برخی اشکالاتی که درشبکه به وجود می آید، اختلالی در ارائه خدمات به وجود نمی آورد، (مانند مسائل QOS) که ردیابی و پیدا کردن این مشکلات بسیار پیچیده خواهد بود. امّا درشبکه ای که در آن MRTG پیاده سازی شده، باتوجه به رفتار Link ها می توان دریافت که کدام Link ها یاDevice ها باتوجه به پیشینه اش، رفتاری غیر عادی از خود نشان می دهد.

مزایای استفاده از MRTG :

۱. باتوجه به ارائه نرم افزار MRTG تحت Public General License هیچ گونه هزینه ای جهت خرید آن به استفاده کننده تحمیل نمی شود .

۲. نصب و راه اندازی آن نسبتا آسان بوده ونیازی به منابع زیاد (از قبیل Process، Disk.RAM) ندارد.

۳. قابل پیاده سازی به زبانهای کشورهای مختلف می باشد .

۴. سازگاری با بسیاری از سیستم عامل ها از قبیل :

Linux 1.2.x, 2.0.x, 2.2.x, 2.4.x (Intel and Alpha and Sparc and PowerPC)
Linux MIPS, Linux S/390
SunOS 4.1.3
MacOS X 10.3 with Fink
Solaris 2.4, 2.5, 2.5.1, 2.6, 7, 8, 9
AIX 4.1.4, 4.2.0.0, 4.3.2
HPUX 9,10,11
WindowsNT 3.51, 4.0, 2k, XP, 2003 (95, 98 and ME too)
IRIX 5.3, 6.2, 6.5
BSDI BSD/OS.۴ ,۲.۱ x, 3.1
NetBSD 1.5.x 1.6.x
FreeBSD 2.1.x, 2.2.x, 3.1, 3.4, 4.x
OpenBSD 2.x, 3.x
Digital Unix 4.0
SCO Open Server 5.0
Reliant UNIX
NeXTStep 3.3
OpenStep 4.2
Mac OS X 10.1 or greater

۵. دارای قابلیت scalability بالا

به تازگی ابزاری کمکی برای MRTG تهیه شده که RRD Tool نام دارد. این نرم افزار با قرارگرفتن درکنارMRTG می تواند Flexibility بیشتری درایجاد Data base های مربوط به گرافها را تامین نماید. به طوری که محدودیت های عنوان شده به گونه ای چشمگیر کاهش می یابد. درحقیقت MRTG با مجهز شدن به برنامه RRD Tool با سرعت وکارایی بالا تری می تواند عمل نماید . هرروز به تعداد استفاده کنندگان نرم افزار MRTG افزوده می شودو سعی می شود نواقص این برنامه مرتفع شود تا بتوان از مزایای زیاد آن بهره برد و تا آنجا که امکان دارد از MRTG جهت کنترل هرگونه تجهیزات شبکه استفاده نمود و این امکان ایجاد شود تا انواع مختلف object ها توسط MRTG قابل Monitoring باشند. در این راستا یک Newsgroup و Mailing List جهت ارائه انواع پرسشها ویافتن پاسخ سوالات مختلف در مورد مشکلاتی که ممکن است با آن مواجه بود ایجاد گردیده است،که آدرس آن به شرح ذیل می باشد.

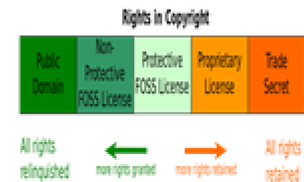
برای دانلود میتوانید به این سایت مراجعه کنید:

http://oss.oetiker.ch/mrtg/download.en.html

@rooham_inet

عهدنامه ها

آیا تا به حال کدی نوشته اید و یا از قطعه کدی که دیگران نوشته اند استفاده کرده اید؟ به احتمال بسیار زیاد جواب شما مثبت است. حتی شاید شما برنامه نویس نبوده باشید اما از برنامه هایی مثل تلگرام و یا گجتهایی مثل تلویزیون های هوشمند، سیستم عامل های مختلف استفاده می کنید با این تفاسیر شما نیازمند دانستن انواع لایسنس ها و تفاوت آنها با یکدیگر هستید. من قصد دارم به صورت بسیار ساده و خلاصه این قوانین و تفاوت آنها با یکدیگر را بررسی کنم اما از دیدگاه پیچیده تر وارد مباحثی مانند مالکیت در مقابل لایسنس و یا حقوق دیدن در مقابل پنهان بودن کد خواهیم شد که از آنها می گذریم. به صورت کلی ما لایسنس ها را به دو دسته ی آزاد- متن باز و اختصاصی تقسیم می کنیم.



نمودار نشان میدهد هرچه به سمت راست حرکت کنیم حقوق کمتری به کاربر داده می شود و هرچه به سمت چپ حرکت کنیم حقوق بیشتری به کاربر داده خواهد شد.

- **نرم افزارهای آزاد متن باز:**

از معروفترین لایسنس های این دسته بندی می توان به **GPL**ها (**GNU General Public License**) اشاره کرد که دارای سه نسخه می باشند. لینوس توروالدز ابتدا نسخه یک آن را وضع کرد و سپس آن را به نسخه ی دوم ارتقا داد اما نسخه سوم را مناسب نمی دانست و آن را دارای پیچیدگی ها حقوقی درنظر گرفت.



بنا بر این کرنل با لایسنس نسخه دوم وجود دارد. حال مفهوم کلی **GPL** چیست؟

این مجموعه لایسنسها بر اساس یک مفهوم خیلی ساده کار می کنند. اگر کدی را بهبود می دهید باید کد جدید را به صورت عمومی نشر دهید (و یا به مالک اولیه کد بدهید). حال این اصل مهم شامل یک تبصره می شود، اگر یک کد تغییر داده شود اما استفاده عمومی نداشته باشد، نیازی به نشر کد نمیباشد.



برای مثال شرکتهایی مثل مایکروسافت و یا گوگل دارای نسخه بهینه شده خود می باشند اما به دلیل اینکه از آن استفاده عمومی نکرده اند پس طبق این تبصره نیازی به منتشر کردن کد جدید ندارند. حال یک لایسنس مشابه و نزدیک به **GPL** هم وجود دارد که تقریبا در بیشتر موراد شبیه یکدیگر هستند اما از کتابخانه های اشتراکی هم استفاده میکنند که با **LGPL** معروف شده است. از سایر مجموعه لایسنس های آزاد و متن باز نیز می توانیم به **MIT** و **BSD** وآپاچی و سایر موارد نام برد که از بین آنها هم **BSD** ها دارای مخاطبان بیشتری هستند پس به



بررسی اجمالی آنها می پردازیم.

فرض کنید شما یک کمپانی هستید و قصد دارید یک قطعه کد را بگیرید و سپس آن را توسعه دهید و یک قطعه کد انحصاری تولید کنید اما آن کد انحصاری را به صورت عمومی نشر دهید برای این امکان شما باید از **BSD** ها استفاده کنید شاید برای غالب افراد عجیب به نظر برسد که چرا این لایسنس وجود دارد اما به هر حال در مواردی خاص شرکتهای بزرگ هم دست به حمایت از مرجع کدها می زنند و با پرداخت مبلغ زیادی به توسعه ی کد اصلی کمک می کنند از معروفترین مثال این مورد می توانیم به قسمتهای زیادی از **OS X** اشاره کرد البته بجز قسمتهای گرافیکی و سایر مواردی که تحت لایسنس **EULA** شرکت اپل هستند.

نرم افزارهای اختصاصی:

حال می رسیم به قسمت سیاه ماجرا برای کاربران احتمالا تا امروز از ویندوز و یا نرم افزارهای تجاری و انحصاری مثل ادوبی فتوشاپ و یا مایکروسافت آفیس استفاده کرده اید. این نرم افزارها بر مبنای مفهوم کپی رایت و لایسنس **EULA** یعنی **End User Agreement** کار می کنند.



حال **EULA** چیست؟ یک قرار داد الکترونیکی است که پس از اینکه آن را قبول کردید خیلی از امکانات از شما گرفته می شود از این موارد می توان به عدم مهندسی معکوس ، عدم دسترسی به سورس کد، عدم تغییر در سورس کد، جمع آوری اطلاعات برای دلایلی مثل بهبود کیفیت سرویس و یا موارد نامعلوم اشاره کرد. این لیست معمولا چند صفحه می باشد و تقریبا هرگونه حق را از کاربر نهایی صلب می کند. خلاصه این که اگر آزاد فکر می کنید کدی که زحمت طراحی آن را خود شما کشیده اید را آزاد کنید تا با سرعت هرچه سریع تر توسعه یابد.

(@AAP1024)

گیک و سلامت

اینبار میخواهیم در مورد تغذیه سالم یک گیک در یک روز صحبت کنیم.

اینکه ما در طول روز فقط ۳ بار وعده اصلی غذا داریم به دوران سزار روم بر میگردد که سعی داشت استانداردی برای زندگی شهری و اجتماعی ساکنین امپراطوری روم به وجود بیاورد.

بدلیل اینکه ساعات خواب هر گیک با گیک دیگر فرق میکند پس از کلماتی مثل (صبحانه – ناهار – عصرانه و شام) در این متن استفاده نمیکنیم بلکه بنا را بر این قرار میدهیم که در زمان بیداری هر ۳ ساعت یک وعده غذایی مصرف می‌شود.

وعده اول : املت پنیر و بروکلی



مدت زمان آماده سازی : ۵ دقیقه

مدت زمان پخت : ۱۰ دقیقه

مقدار کالری دریافتی : ۲۹۰ واحد

مواد لازم:

– ۲۰۰ گرم کلم بروکلی

– ۲ عدد تخم مرغ بزرگ

– ۲ قاشق غذاخوری پنیر فتا

– ۲ قاشق غذاخوری روغن

– نصف قاشق چایخوری نمک

– نصف قاشق چایخوری فلفل

روش پخت :

ابتدا روغن را در یک ماهیتابه نچسب میریزیم و صبر میکنیم تا گرم شود و سپس کلم بروکلی را خرد کرده و برای ۳ دقیقه در روغن تفت میدهیم.

در یک کاسه تخم مرغ و پنیر خرد شده را با هم مخلوط میکنیم و نمک و فلفل را به آن اضافه میکنیم. این مخلوط را به کلم ها اضافه میکنیم و بین ۳ تا ۴ دقیقه اجازه میدهیم پخته شود و سپس املت را در داخل ماهیتابه برمیگردانیم و به مدت ۲ دقیقه اجازه میدهیم تا سمت دیگر آن پخته شود و سپس آن را با نان تست سرو میکنیم.

وعده دوم: ساندویچ موز و شکلات



مدت زمان آماده سازی : ۵ دقیقه

مقدار کالری دریافتی : ۲۰۰ واحد

مواد لازم:

– ۲ قاشق سوپ خوری شکلات صبحانه

– ۱ موز متوسط

– ۲ عدد نان تست شده

روش تهیه :

شکلات را روی نانهای تست شده بمالید و سپس موز را بصورت حلقه حلقه برش داده و روی یکی از نانها بچینید و سپس نان دوم را روی موزها قرار دهید.

وعده سوم : خوراک لوبیا



مدت زمان آماده سازی : ۱۵ دقیقه

مدت زمان پخت : ۱۰ دقیقه

مواد لازم:

–۲ قاشق چایخوری روغن زیتون

–۱ عدد پیاز بزرگ خرد شده

–۱ عدد فلفل سبز کوچک خرد شده

–۲ حبه سیر خرد شده

–۱ قاشق چایخوری فلفل قرمز

–۱ عدد گوجه متوسط خرد شده

–۱ عدد کنسرو لوبیا سیاه

–۲ قاشق ماست پرچرب

روش پخت :

در یک ماهیتابه روغن را داغ میکنیم و سپس پیاز و فلفل سبز را به مدت ۳ دقیقه با هم تفت میدهیم بعد سیر و فلفل قرمز را اضافه میکنیم و به مدت ۲ دقیقه تفت میدهیم بعد گوجه ها و لوبیا را به مخلوط اضافه میکنیم و به مدت ۵ دقیقه با بقیه مواد میپزیم.

در زمان سرو ۲ قاشق ماست پرچرب را روی خوراک اضافه میکنیم و برای مزه بهتر میتوانید از لیمو تازه یا آبلیمو استفاده نمایید.

وعده چهارم: سالاد ماکارونی



مدت زمان آماده سازی : ۳۰ دقیقه

مدت زمان پخت : ۱۵ دقیقه

مقدار کالُری دریافتی : ۱۰۰۰ واحد

مواد لازم :

– ماکارونی ۴۰۰ گرم

– فلفل دلمه ۱ عدد

– گوجه ۲ عدد

– کالباس ۳۰۰ گرم

– خیارشور ۲۰۰ گرم

– کنسرو ذرت ۱ عدد

– سس مایونز ۳ قاشق

– ۱ قاشق نمک

روش پخت:

نصف یک قابلمه متوسط را با آب پرکنید و به همراه نمک بگذارید تا جوش بیاید. در همان زمان کالباسها و گوجه ها و فلفل دلمه و خیار شور ها را خرد کنید وقتی آب جوش آمد ماکارونی را اضافه کنید و اجازه دهید کاملاً پخته شوند و سپس آن‌ها را آبکش کنید و بعد تمام مواد خرد شده را به آن اضافه کنید و بهم بزنید ذرت را هم اضافه کنید و بعد سس مایونز را اضافه کنید و خوب بهم بزنید، بعد به مدت ۱۵ دقیقه در یخچال قرار دهید تا سرد شود.در زمان سرو میتوانید کمی کچاپ به آن اضافه نموده و میل نمائید.

این وعده غذایی بدلیل داشتن نشاسته فراوان و همچنین استفاده از سس مایونز که ماده اولیه آن تخم مرغ میباشد از میزان نشاسته و پروتئین بالا برخوردار است که بهتر است در آخر روز یا قبل از خواب مصرف نشود.

وعده پنجم : خوراک سینه مرغ



مدت زمان آماده سازی ۱۵ دقیقه

مدت زمان پخت : ۴۰ دقیقه

مقدار کالُری دریافتی : ۸۰۰ واحد

مواد لازم :

– سینه مرغ کامل۱ عدد

– روغن ۴ قاشق

– سیب زمینی متوسط ۲ عدد

روش پخت:

سینه مرغ را به همراه ۴/۱ پیمانه آب و ۱ قاشق روغن به همراه نمک و فلفل در یک ظرف گود نسوز قرار داده و درطبقه زیرین فر و با حداکثر درجه دما برای مدت ۴۰ دقیقه بپزید.

در همین زمان سیب زمینیها را شسته و خرد کرده و در ۳ قاشق روغن به همراه مقداری نمک سرخ نمایید.

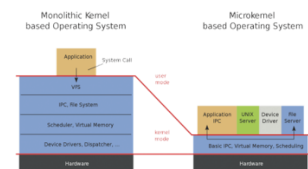
در پایان ظرف را به طبقه بالای فر منتقل کنید و برای ۱۰دقیقه اجازه دهید برشته شود. در زمان سرو میتوانید از سس تند به همراه کچاپ استفاده نمایید.

امیدوارم از تهیه و میل کردن این وعده‌های غذایی لذت برده باشید.

(@BoBzBoBz)

شکست، پدر پیروزی

پروژه گنو با ساخت یک جعبه ابزار کامل شروع شد در واقع خلق منظومه ای از نرم افزار های مورد نیاز برای یک سیستم عامل کامل و صد البته آزاد اما بدون هسته (kernel) اصلی که بتوان با استفاده از آن یک سیستم عامل کامل داشت. بعد از چند پیشنهاد همچون TRIX ، BSD و بحث هایی که شکل گرفت قرار بر این شد که هسته ای تحت عنوان GnuHurd را برای پروژه ی گنو طراحی و پیاده سازی کنند، اما درباره طراحی Hurd، برای اینکه انعطاف پذیری و سرعت را بتوان بالا برد از روشی به عنوان ریز هسته یا همان microkernel استفاده شد. microkernel در واقع یک نوع معماری برای طراحی هسته ی سیستم عامل است که بر خلاف روش سنتی (monolithic) هسته را به چند قطعه و برنامه های کوچکتر تقسیم میکند که با پیام های ناهمزمان (Asynchronous messaging) با هم ارتباط برقرار میکنند .



برخی ویژگی های معماری GnuHurd :

همانطور که درمتن فوق ذکر شد GnuHurd برخلاف لینوکس و تمام هسته های شبه یونیکس از معماری کارخواه/کارساز (Client-Server) که در ریزهسته است استفاده می‌کند و مسئولیت ارائه اکثر سرویسهای هسته و هماهنگی دسترسی به سخت‌افزار رایانه از جمله : CPU (از طریق مدیریت فرایند، زمان‌بندی)، حافظه (از طریق مدیریت حافظه)، و دیگر دستگاه‌های ورودی/خروجی دیگر (از طریق زمان‌بند ورودی/خروجی) برای صدا، گرافیک، ذخیره سازی انبوه و... را فراهم می‌کند. در ابتدای کار، GnuHurd برای استفاده از Gnu Mach به عنوان ریزهسته توسعه داده می‌شد. این یک تصمیم فنی بود که به وسیله ریچارد استالمن گرفته شد، که فکر می‌کرد به دلیل صرفه‌جویی در کار، سرعت توسعه را افزایش خواهد داد اما او اذعان کرد که این تصمیم او اشتباه بود، زیرا در ابتدای کار، درایوهای فضای کاربر به دلیل سربار ارتباطات بین فرآیندی در هسته عملکرد مناسبی نخواهند داشت و عیب یابی این ارتباطات فرآیندی کاری بس دشوار و زمانگیر بود. البته با کمک عملکرد ماشین‌های امروزی، ممکن است که این سربار باعث مشکل قابل توجهی نشود اما هنوز مشکل ناپایداری و بروز خطاهای فیکس نشده ارثیه ی همان تصمیم فنی است.

اما امتیازاتی که استفاده از microkernel برای GnuHurd میتواند داشته باشد:

۱ - جدایی خدمات (هسته ها) دارای این مزیت است که اگر یکی از این سرویس ها با شکست یا مشکلی مواجه شود دیگرسرویس ها می توانند به کار خود ادامه دهند، بنابراین قابلیت اطمینان از ویژگی های اصلی است که میتوان برای این نوع از معماری ذکر نمود. بر خلاف هسته هایی که با معماری یکپارچه توسعه داده شده اند اگر در بخشی از یک سرویس ریز هسته ها مشکلی به وجود بیاید بقیه ی ریز هسته ها میتوانند به کار خود ادامه دهند همچنین این خاصیت موجب نگهداری راحت تر از هسته نیز میشود.

۲ - خدمات مختلف به صورت ماژول هایی هستند که در زمان اجرا می توانند لود (بار) شده و در زمانی که مورد نیاز باشد از رم سیستم خارج(تخلیه) شوند. در هنگام تست سیستم آن قطعه ی خاص میتواند مورد بررسی قرار گیرد و پس از انجام عملیات های لازم در جای خود قرار گیرد.

۳ – رد و بدل کردن پیام ها به صورت ارتباطات مستقل موجبات توسعه پذیری بیشتر را فراهم میکند .

۴ – این حقیقت که شما پس از تست یا توسعه ی یک یا چند ماژول نیازی به راه اندازی مجدد سیستم (reboot) ندارید یکی از امتیازات ویژه استفاده از این معماری می باشد .



انتخاب microkernel علاوه بر این امتیازاتی که برای GnuHurd داشت باعث بروز مشکلاتی در این سیستم عامل شد به طوری که با گذشت زمان و با زیاد شدن حجم کد ها کم کم مدیریت خطا و یافتن آنها کار سختی شد به طوریکه مشکلات بزرگی را برای این پروژه ایجاد کرد و تیم توسعه ی این محصول دریافتند که این نوع از معماری (microkernel) نیاز به وقت و هزینه ی بالایی برای کانفیک کردن بخش های مختلف و پیدا کردن مشکلات دارد، به طوری که به گفته ریچارد استالمن "یافتن خطا در سیستمی که در آن پیام های غیر همزمان با ترتیب ها و وابستگی هایی از هم کار میکنند کار بسیار دشواری است و نتیجه آن شد که سال ها طول کشید که توانستیم برنامه را به کار بیندازیم"

همین مشکلات باعث کاهش پایداری GnuHurd و افزایش هزینه و از دست رفتن زمان برای تیم Gnu شد و هسته GnuHurd نتوانست در میان کاربران Gnu همچون لینوکس جایی باز کند . این در حالی بود که در همان سالها تروالدز با استفاده از معماری monolithic یا همان طراحی یک پارچه هسته ای با سرعت بالا و پایداری بیشتر پیاده سازی کرد و جای خالی نبود هسته اصلی را در پروژه Gnu پر نمود.

(@Amrkamankesh)

